# Oracle DataGuard 10gR2

# PSOUG Education

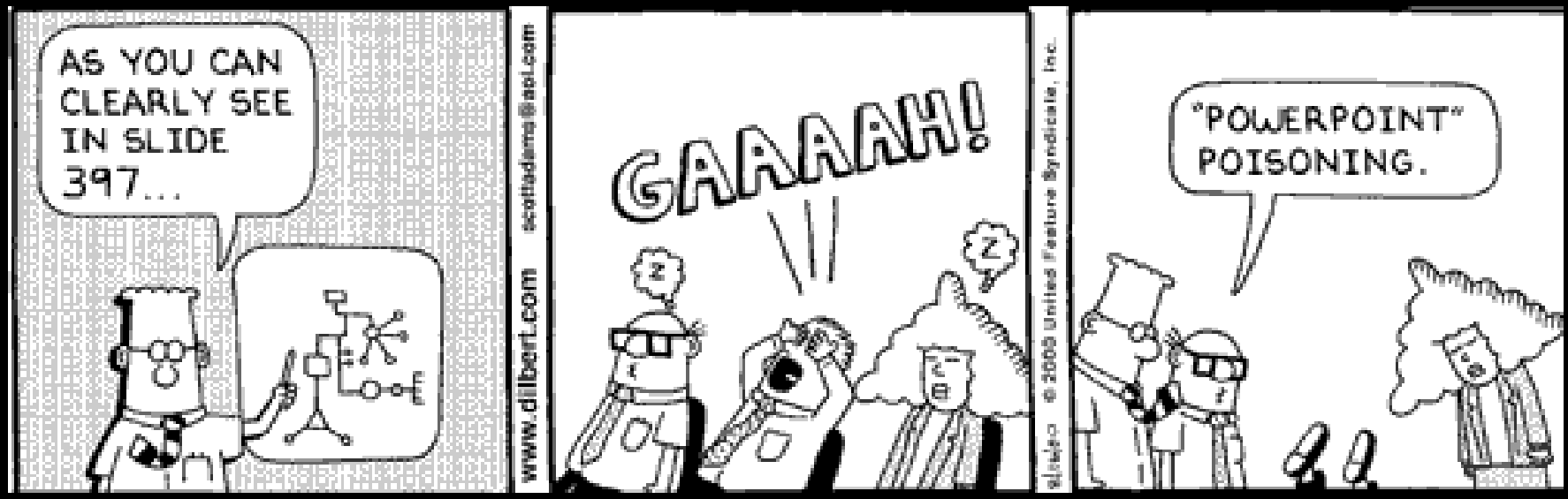
Education Is Our Passion

Hands-on Workshop Series

# Daniel A. Morgan

damorgan@u.washington.edu
206-669-2949

# We Do Not Practice:



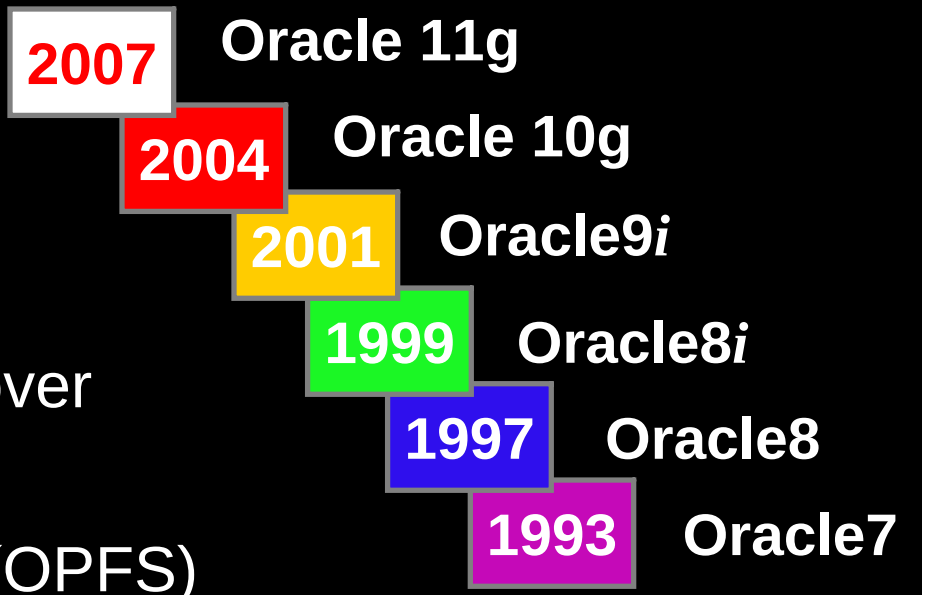

Slow Death by PowerPoint

# Discussion

## Oracle Data Guard Overview

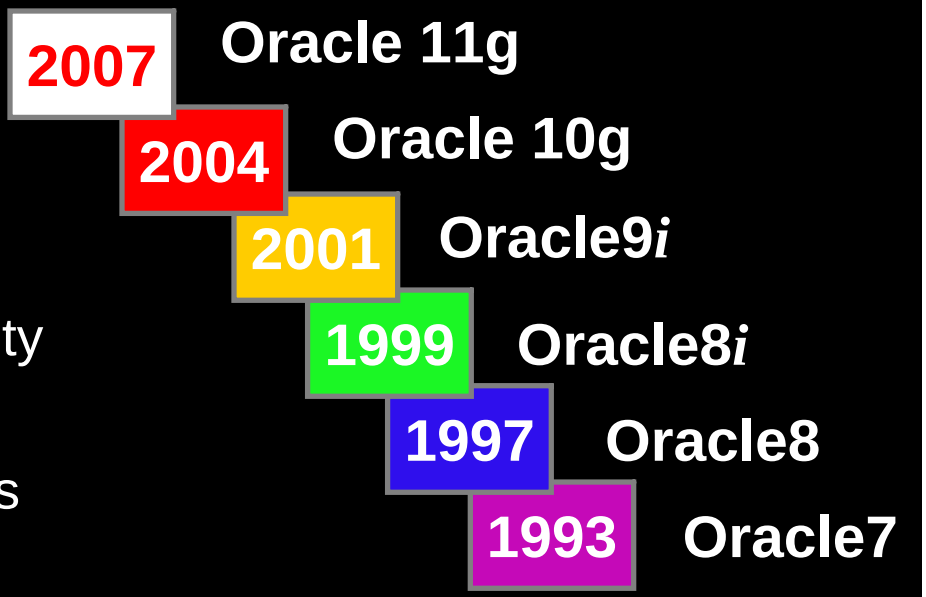
# Data Guard History

- 7.3: Custom Standby Database
- 8i: Automated Standby
  - read-only database
  - managed recovery
  - remote archiving
- 8i: Data Guard
  - Automation
  - Single command switch-over
  - Single command fail-over
  - Oracle Parallel Fail Safe (OPFS)



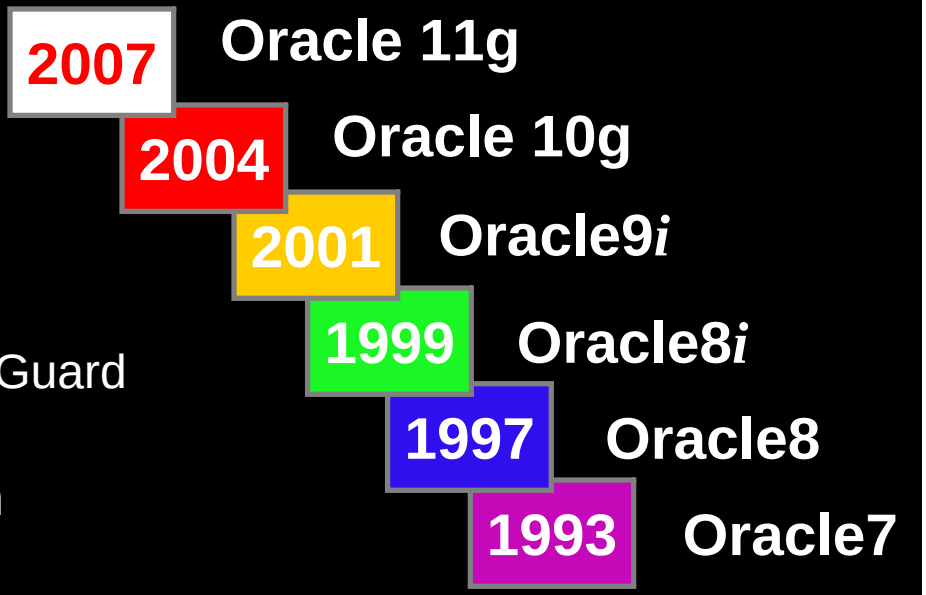
# Data Guard History

- 9iR1: Data Guard
  - Integrated zero-data-loss capability
  - Data Guard Broker w/ Data Guard Manager GUI
  - Command Line Interface (CLI)
  - Switchover and Failover operations
  - Automatic gap resolution
  - Automatic Synchronization
- 9iR2: Data Guard
  - Logical standby databases
  - Maximum protection / availability
  - Enhanced Data Guard Broker
  - Cascaded redo log destinations



# Data Guard History

- 10gR1: Data Guard
  - Real-time apply
  - Recovery through **`OPEN RESETLOGS`**
  - Simplified configuration with **`VALID_FOR`** attribute
  - Standby redo log support on logical standby databases
  - Improved redo transmission security
  - Improved support for RAC
  - Zero downtime instantiation of logical standby databases
- 10gR2: Data Guard
  - Fast-start Failover
  - Flashback Database across Data Guard switchovers
  - Asynchronous Redo Transmission
  - Faster Redo Apply failover



# Fast Start Failover

- Fast-start failover: Automatically, quickly, and reliably fail over to a designated, synchronized standby database
- After a fast-start failover occurs, the old primary database is automatically reconfigured as a new standby database upon reconnection to the configuration.
- Maintain uptime and increase the availability, as well as the robustness of disaster recovery.
- Less need for manual intervention.

# Flashback Database across Switchovers

- It is now possible to flash back the primary and standby databases to an SCN or a point in time prior to a switchover operation. When you use this feature of Flashback Database on a physical standby database, the standby role is preserved. On a logical standby database, the role of the standby database is changed to what it was at the target SCN or time.

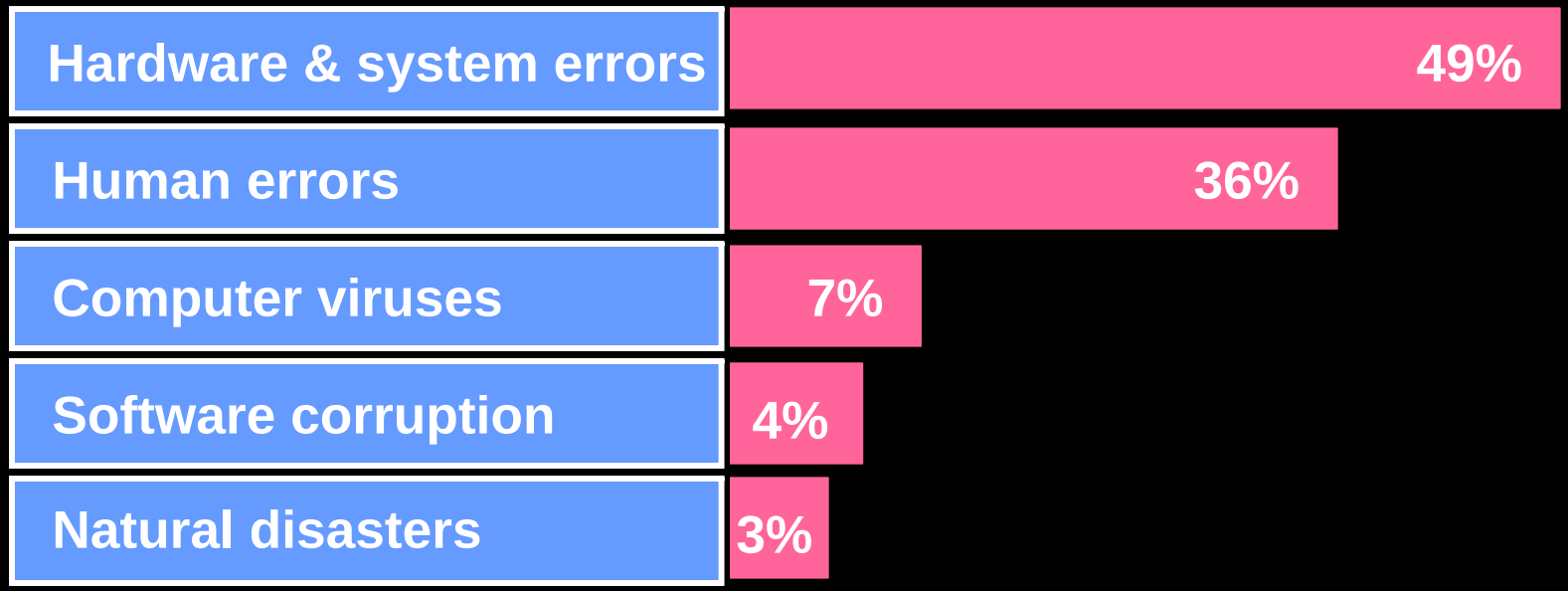
# Asynchronous Redo Transmission

- Asynchronous redo transmission using the log writer process (LGWR ASYNC) has been improved to reduce the performance impact on the primary database. During asynchronous redo transmission, the network server (LNSn) process transmits redo data out of the online redo log files on the primary database and no longer interacts directly with the log writer process.
- Allows the log writer process to write redo data to the current online redo log file and continue processing the next request without waiting for interprocess communication or network I/O to complete.

# Discussion

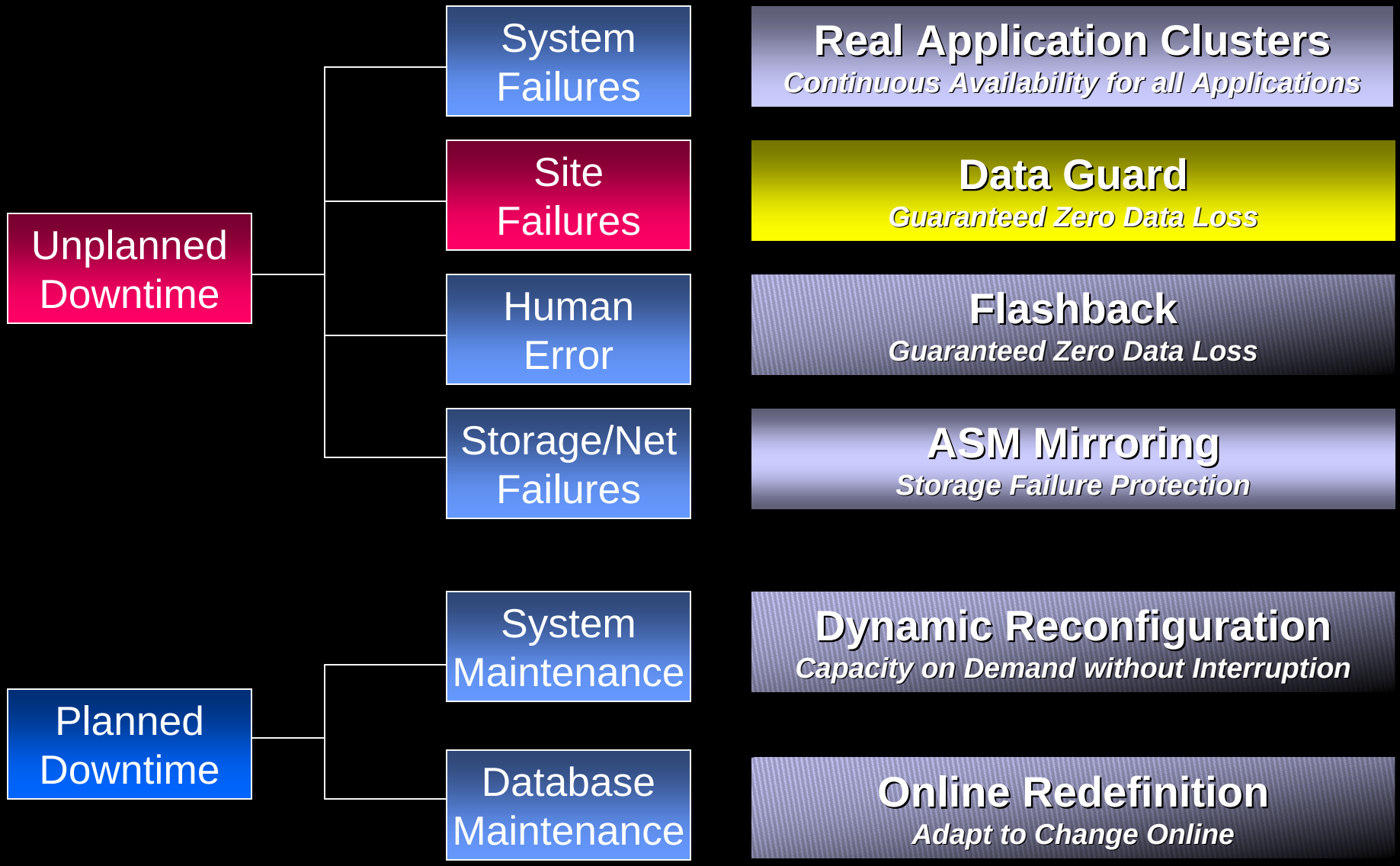
## Data Guard Concepts

# Causes of Data Loss




**Source: *Disaster Recovery Journal***

# Fault Tolerance



# Data Guard Acronyms

- ARC0 Archiver
- DMON Data Guard Monitor
- FAL Fetch Archive Log
- LCR: Logical Change Record
- LGWR Log Writer backend process
- LSP: Logical Standby Process **aka SQL Apply (logical)**
- MRP Managed Recovery Process
  **aka Redo Apply (physical)**
- RFS Remote File Server **(receives undo)**
- TAF Transparent Application Failover

# A Complete High-Availability Solution

- Recovery Manager (RMAN)
- Flashback Database / Query / Table
- Streams Replication
- Grid Control
- Real Application Clusters (RAC)
- Data Guard

Data Guard is one part of a complete HA solution

# What is Data Guard?

- Real-time replication
- Primarily intended for replication between data centers
- Utilizes Log Miner to read redo logs
- Ships redo to remote site via Advanced Queuing
- Most often applies changes to a standby database
- Can cascade changes from standby-to-standby

- Built by the same team that wrote RMAN, Streams, and Change Data Capture

# Why Data Guard?

- High Availability (HA) Disaster Preparation
  - Replication between geographically separated data centers
- Transparent Application Failover (TAF)
- Remove backup process overhead from primary production systems
- Rolling hardware and operating system upgrades and system maintenance
- Rolling database upgrades and patching
- Production mirror for reporting
- System migration (for example to ASM)

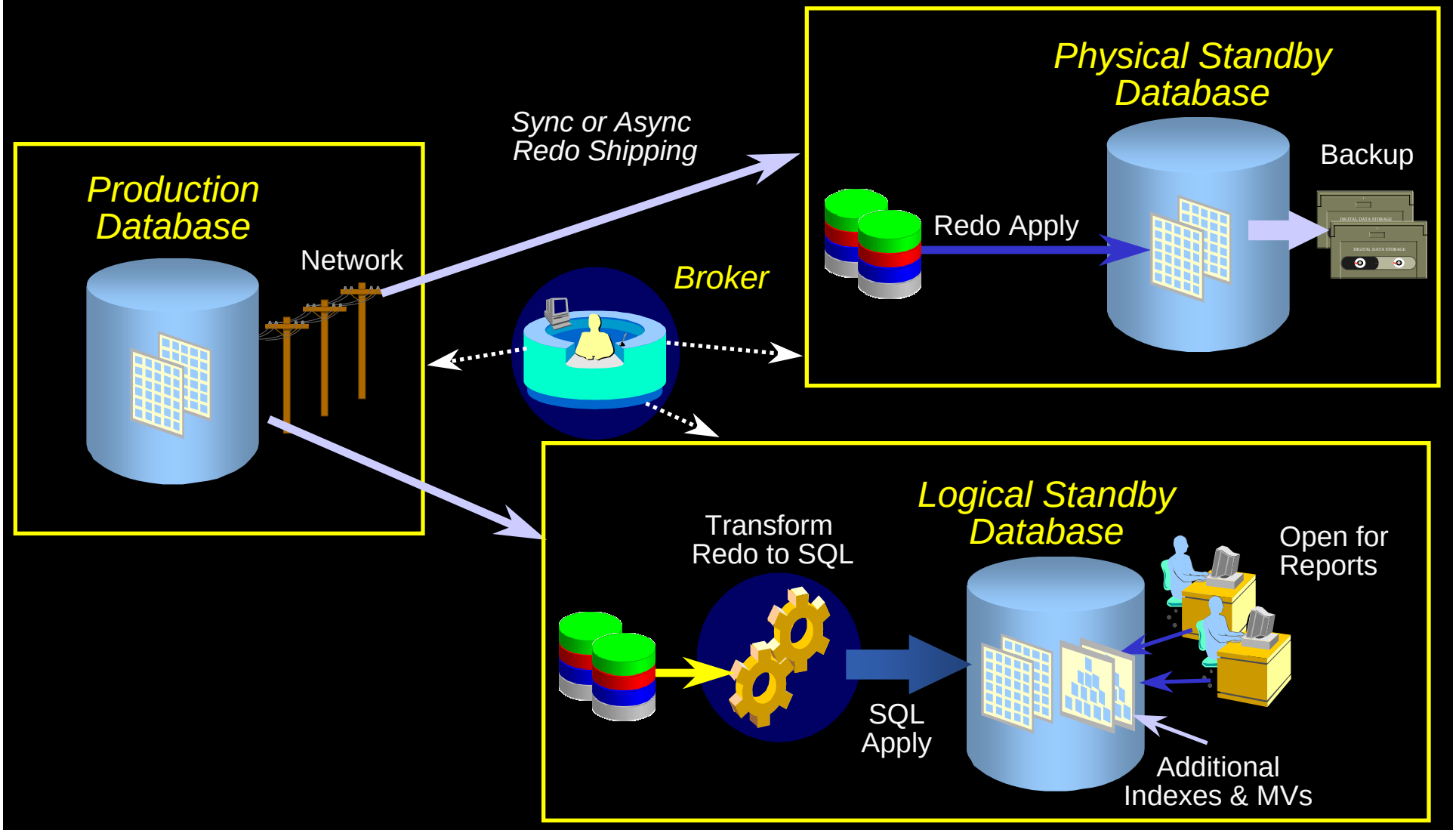
# What is a Standby Database

- A copy of a production database that you can use for disaster protection. You can update the standby database with archived redo logs from the production database in order to keep it current. If a disaster destroys the production database, you can activate the standby database and make it the new production database.
- You can maintain the standby data in one of the following modes:
  - For physical standby databases
    - Redo Apply
    - Open read-only mode
  - For logical standby databases
    - Open read/write mode
- A Standby Database is NOT Data Guard

# Types of Standby Databases

- There are two types of standby databases
  - Physical standby database
    - block-for-block identical with the primary database
    - Synchronized with the primary database by application of redo data
  - Logical standby database
    - Shares the same schema definition
    - Synchronized with the primary database by transforming the data in the redo received from the primary database into SQL statements that are executed

# Oracle Data Guard Architecture



# Database Protection Modes

- Maximum Protection
  - No Data Loss and No data divergence
  - Arch_dest: mandatory, lgwr, sync, affirm
  - Primary db shutdown when unable to access stdby
- Maximum Availability
  - Arch_dest: mandatory, lgwr, sync, affirm
  - Protection auto lowered when stdby is unavailable
- Maximum Performance
  - Arch_dest: lgwr/arch, sync/async, mandatory/optional
  - Minimal performance impact

# Protection Modes and Requirements

| Protection Mode | Redo Transport | Standby Redo Log Files Needed? | Used with Fast-Start Failover? |
|---|---|---|---|
| MAXPROTECTION | SYNC | Yes | No |
| MAXAVAILABILITY | SYNC | Yes | Yes |
| MAXPERFORMANCE | ASYNC or SYNC | Yes for ASYNC | No |

# Data Guard Services

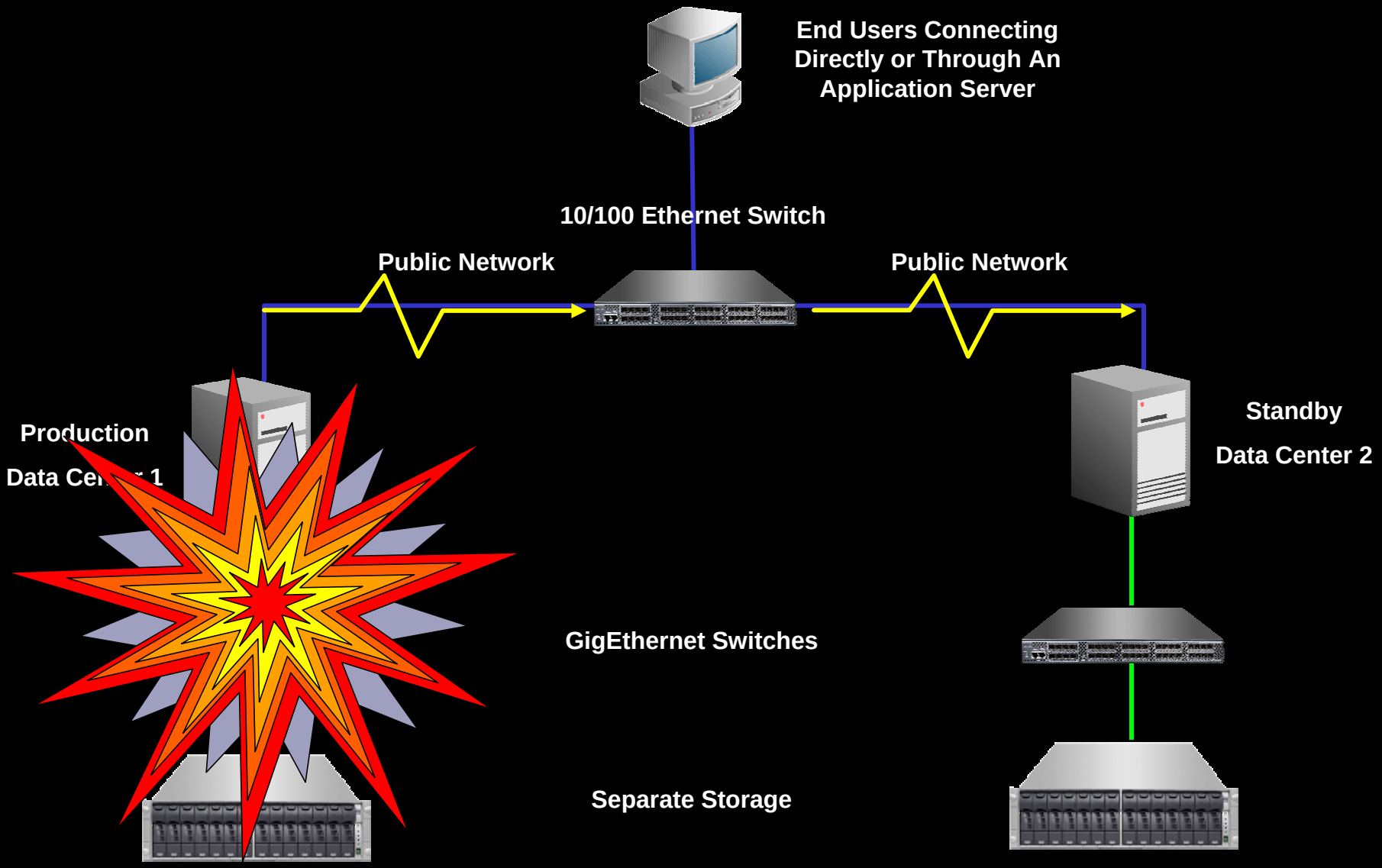
- Log transport services
  - Configurable using ARCH or LGWR
  - Used by Data Guard and Streams
- Log apply services
  - Redo Apply (Physical Standby)
    - Recovers the redo data received from the primary database and applies the redo to the physical standby database.
  - SQL Apply (Logical Standby)
    - Transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database.
- Role-management services
  - what do it do?

# Discussion

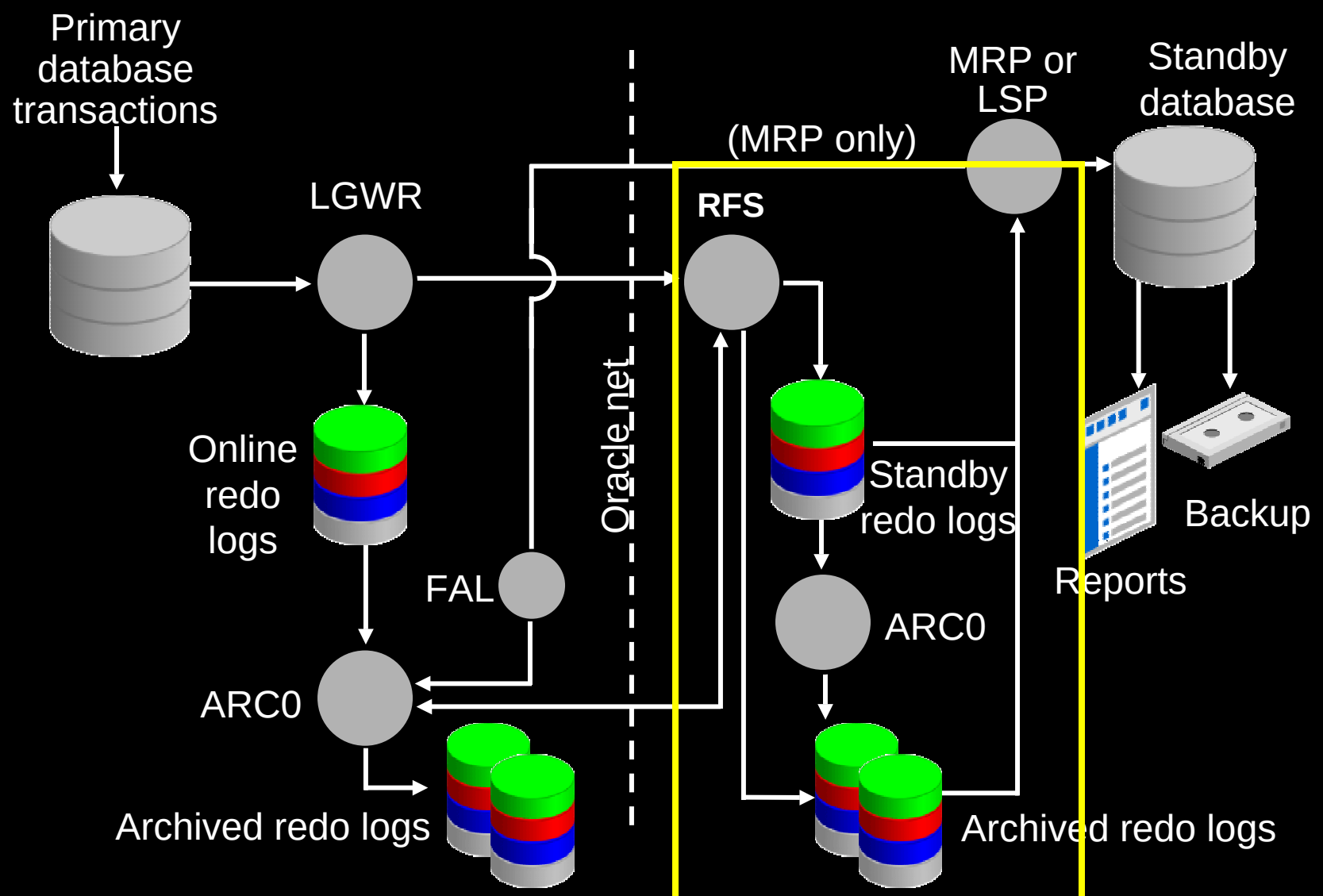
## Data Guard Architecture

## Physical Standby

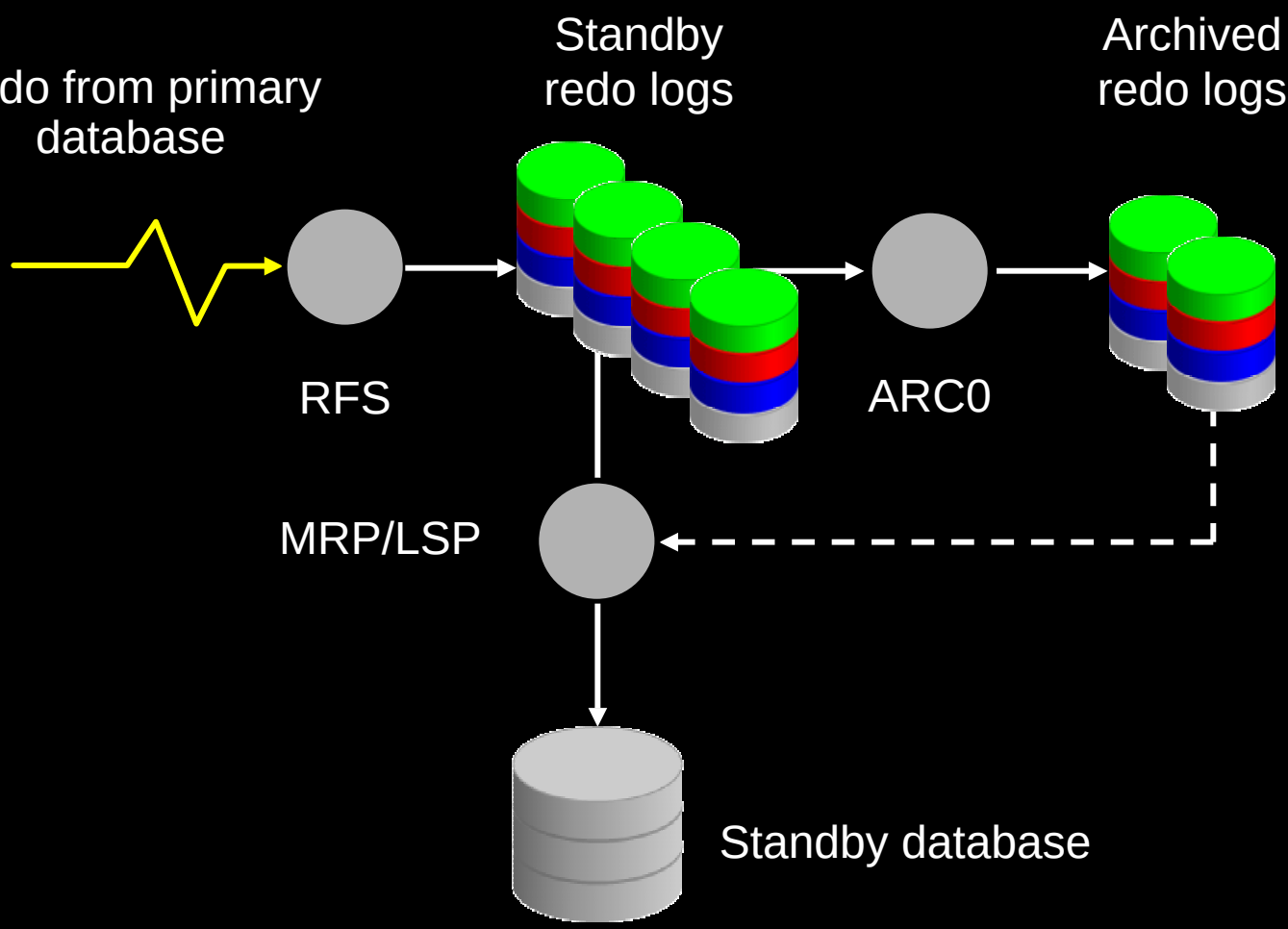
# PSOUG Lab Topology



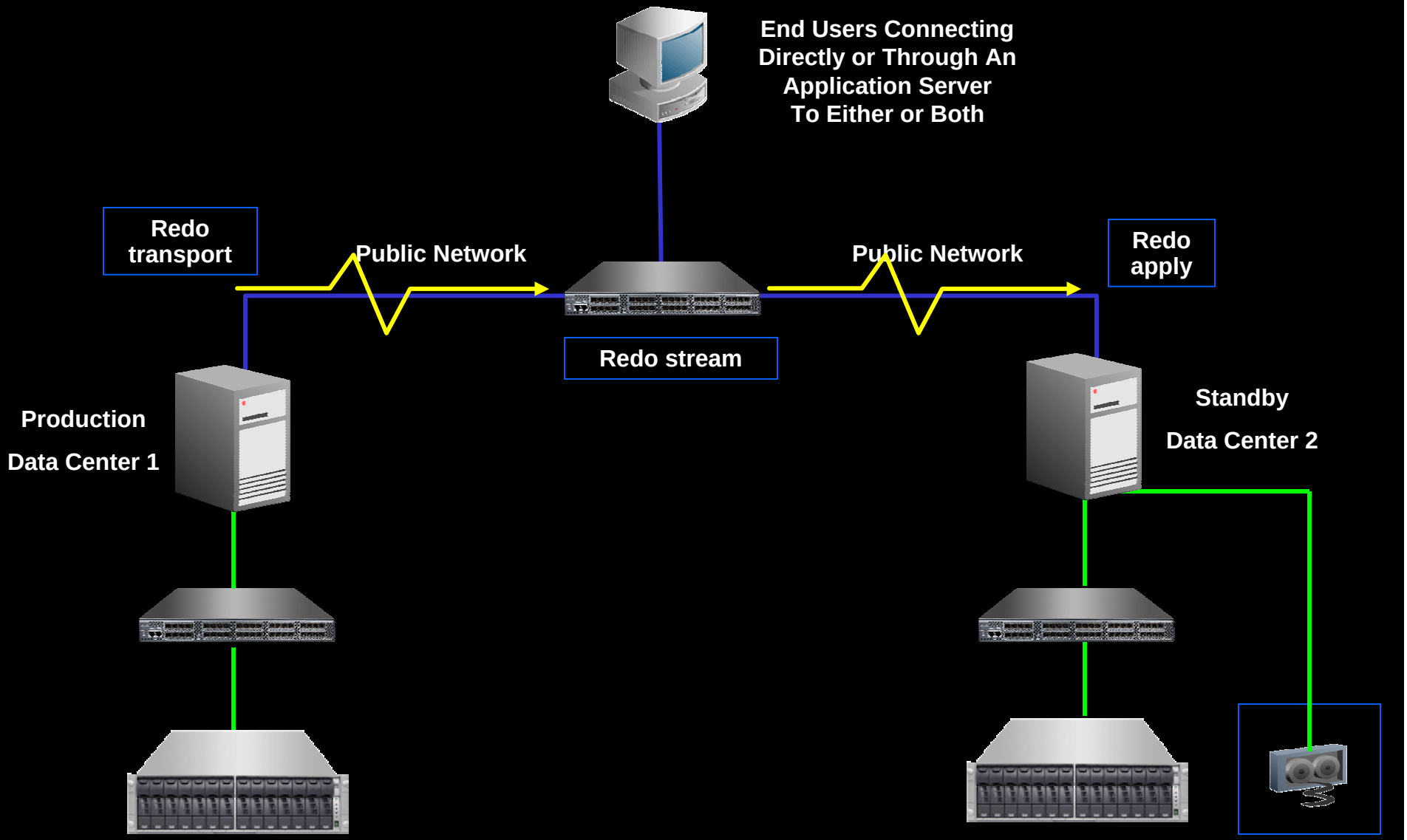
# Oracle Data Guard: Architecture



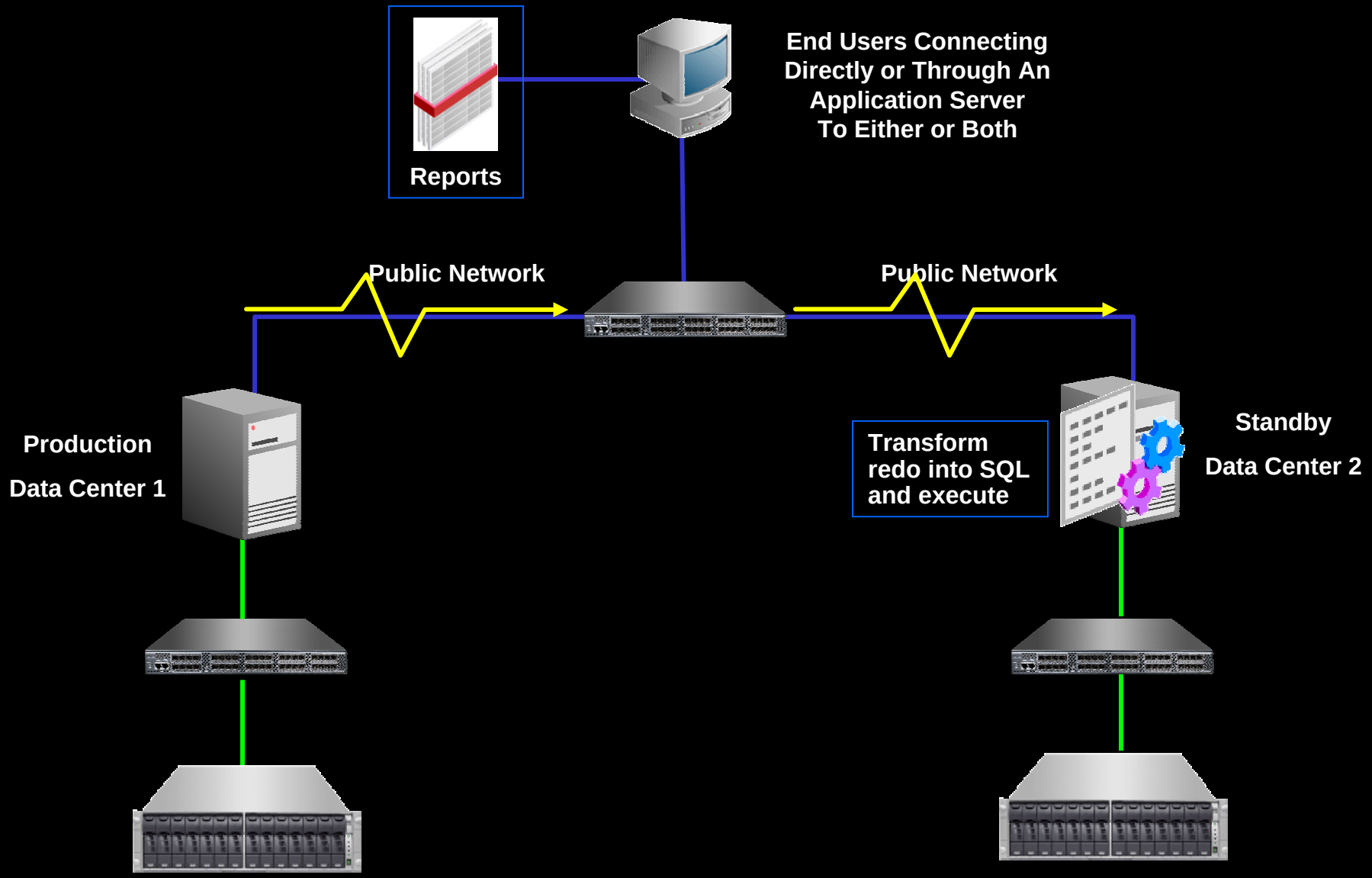
# Standby Redo Logs



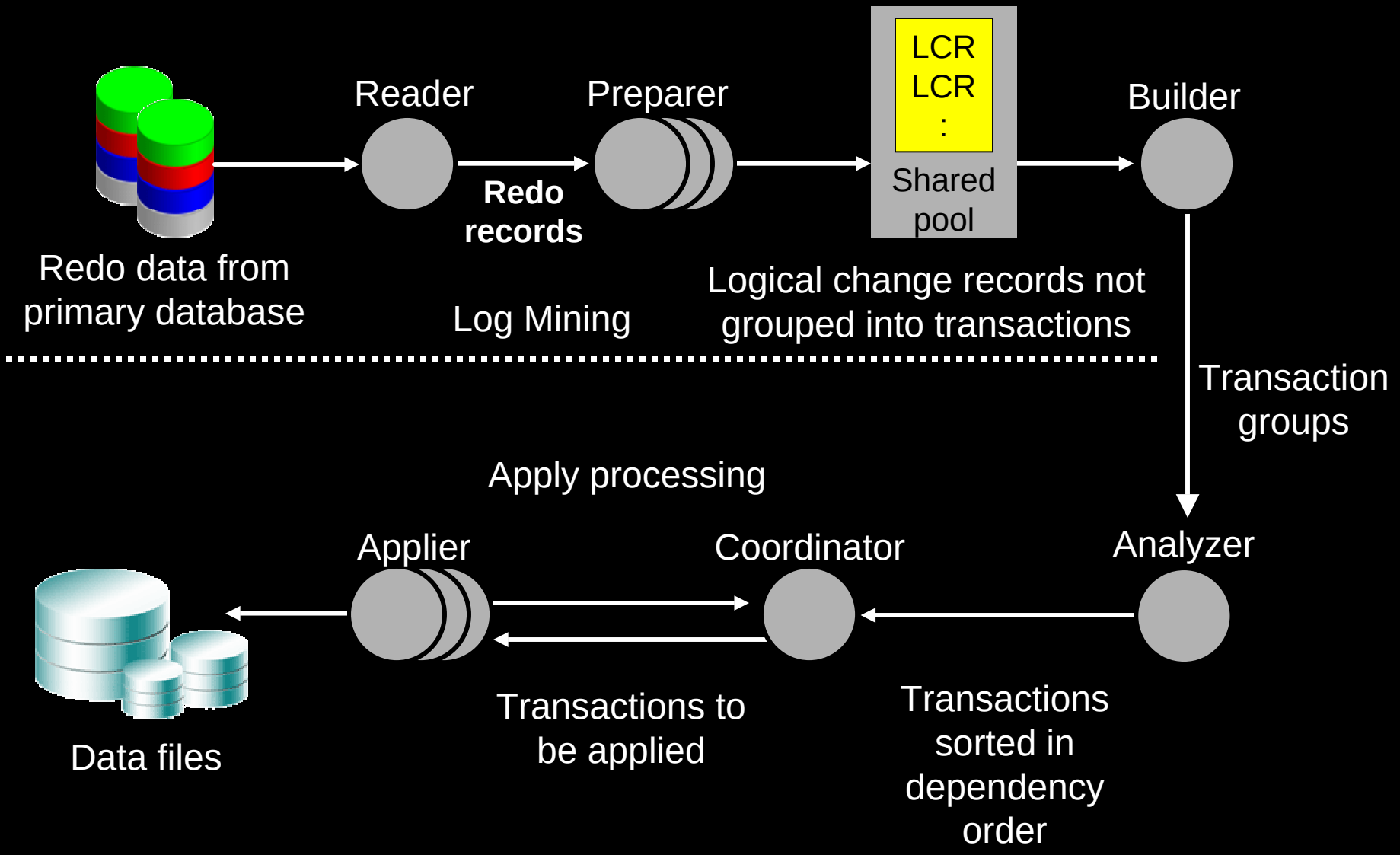
# Physical Standby Database



# Logical Standby Database



# SQL Apply Process: Architecture



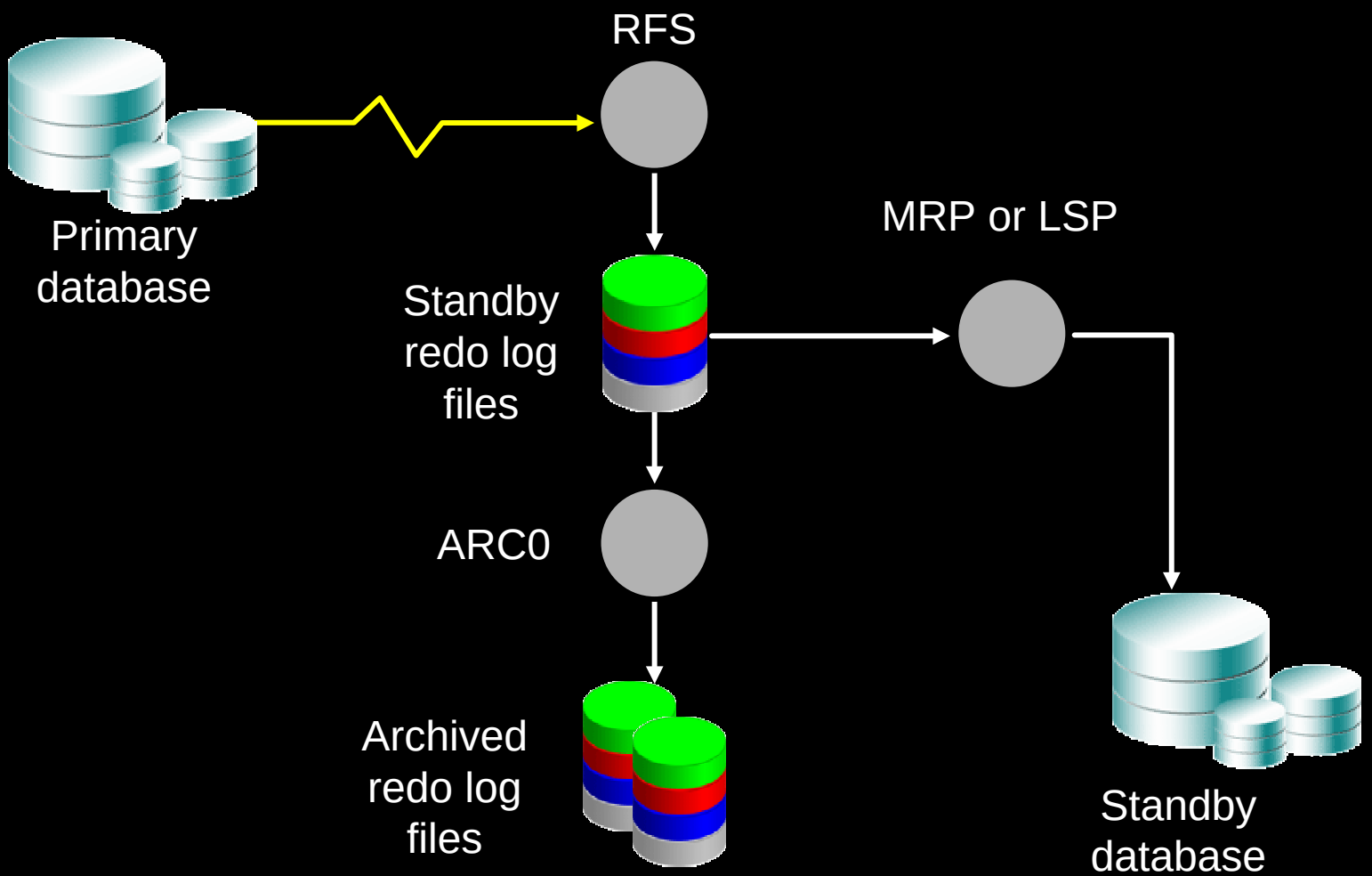
# Real Time Apply

- Redo data is applied to the standby database as soon as it is received from the primary database
  - In Oracle9*i* Data Guard this apply has to wait till an archivelog is created on the standby database
- For Redo Apply:

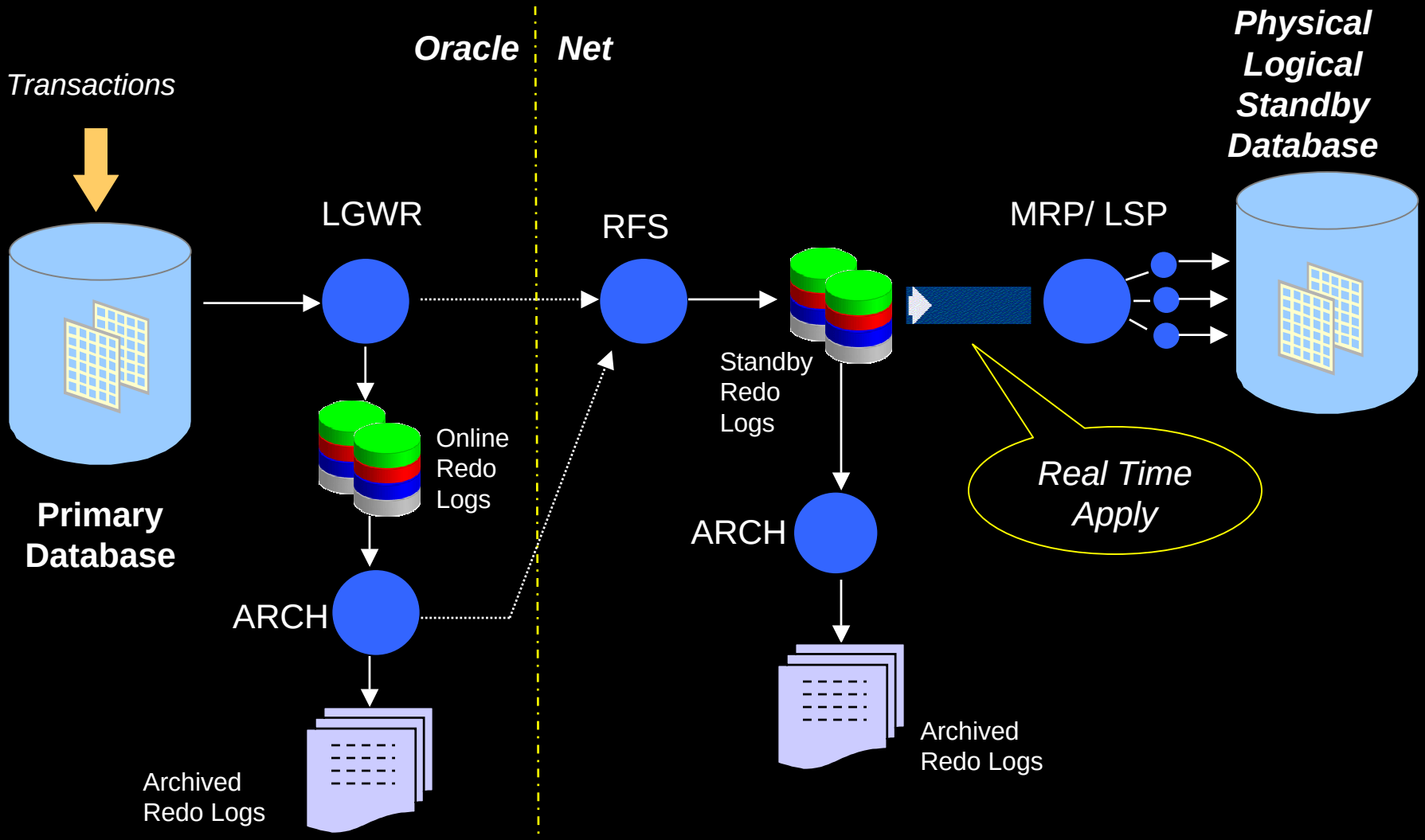
```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE
  USING CURRENT LOGFILE
```

- For SQL Apply:
  `ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE`
- When real time apply is enabled, `RECOVERY_MODE` column in `V$ARCHIVE_DEST_STATUS` displays "`MANAGED REAL TIME APPLY`"

# Real-Time Apply



# Real-Time Apply Architecture



# Real Time Apply – Benefits

- Standby databases now more closely synchronized with the primary
  - More up-to-date, real-time reporting
  - Faster switchover and failover times
    - Reduces planned and unplanned downtime
    - Better Recovery Time Objective (RTO) for DR

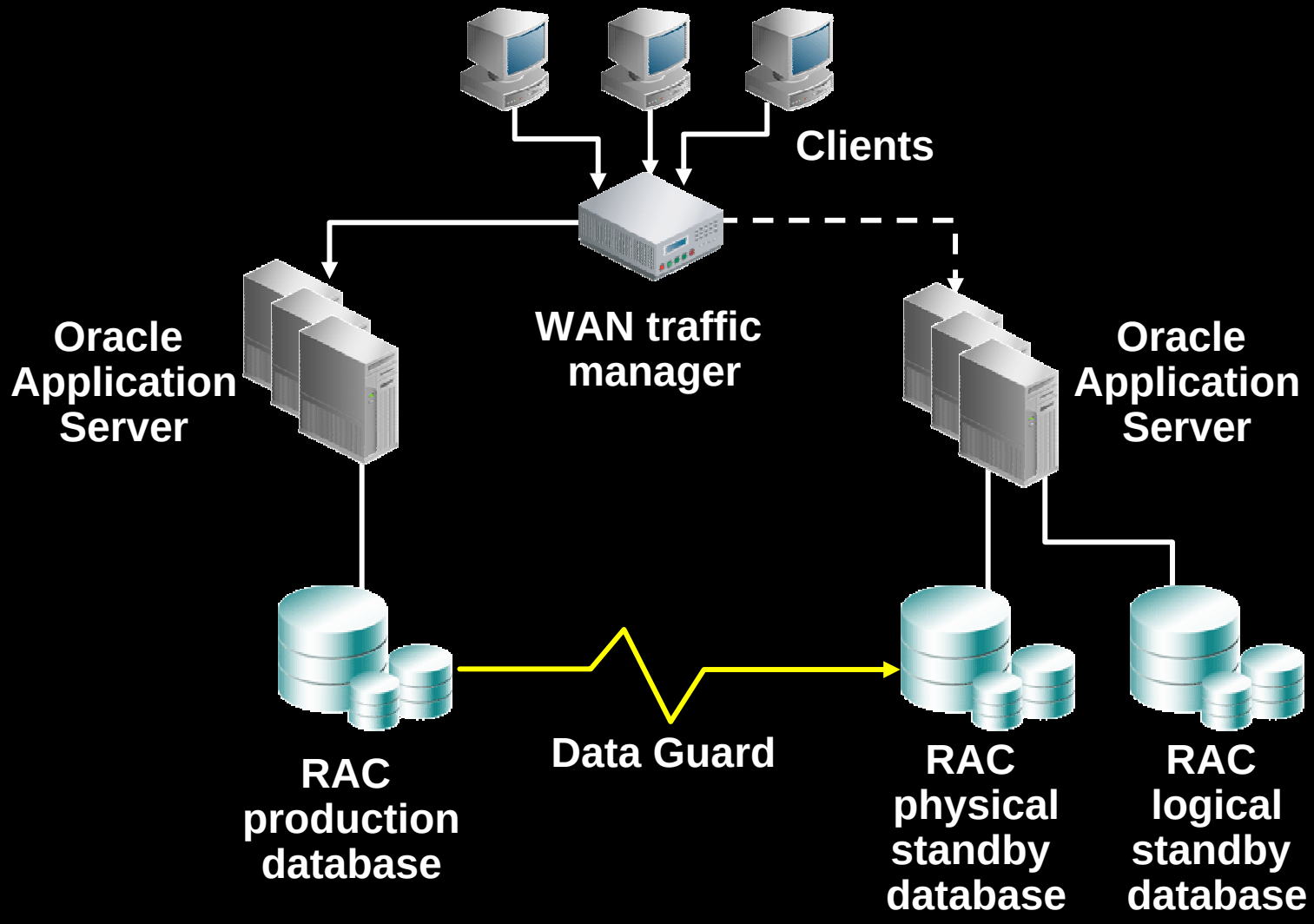
# Discussion

## Data Guard Configurations

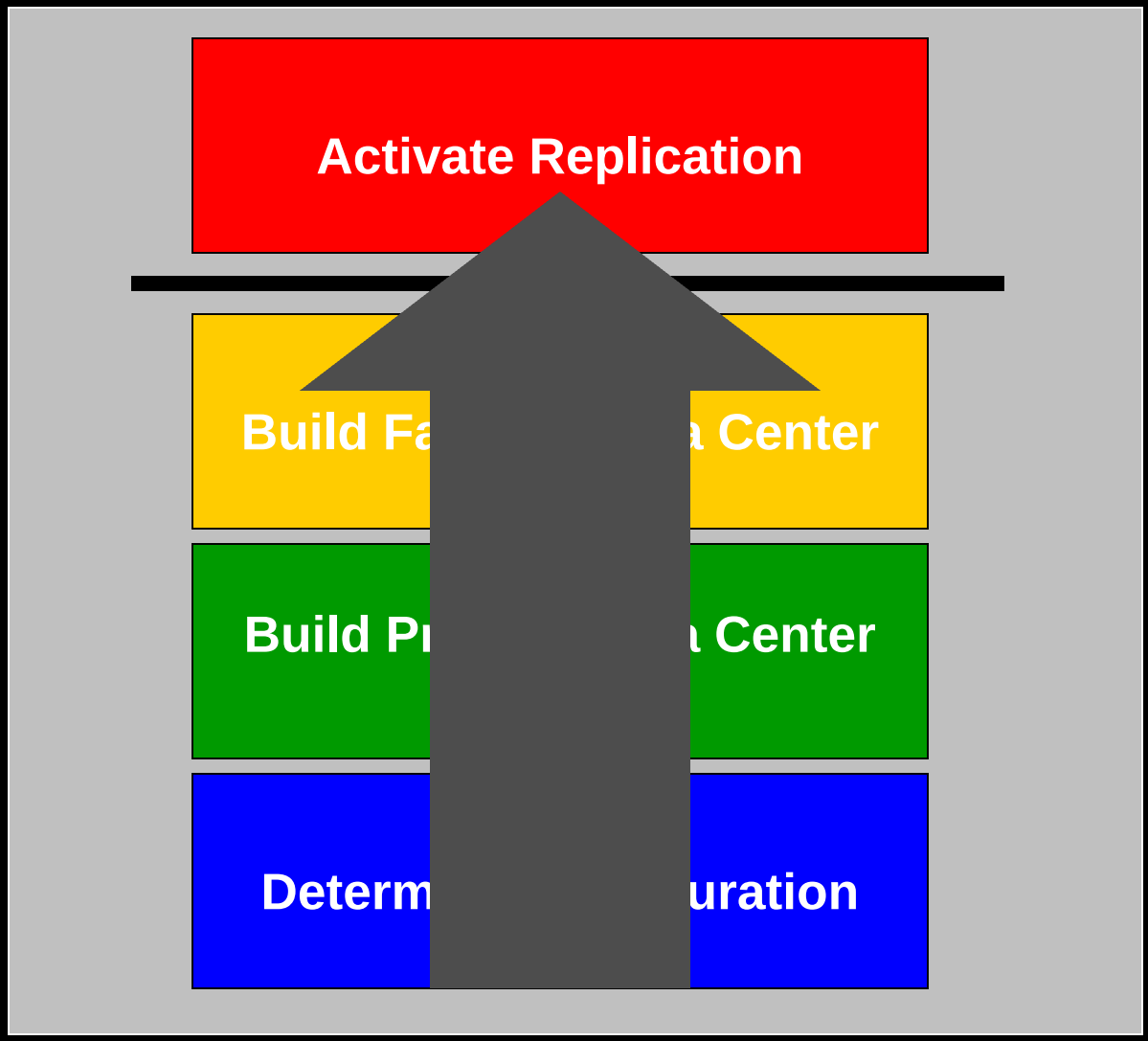
# Data Guard Role Transitions

- Switchover
  - Planned role reversal
  - Used for OS or hardware maintenance
- Failover
  - Unplanned role reversal
  - Use in emergency
  - Zero or minimal data loss depending on choice of data protection mode
- Role-transition operations are not automatically invoked

# Maximum Availability Architecture



# The “Black Line”



# Oracle Data Dictionary (1 of 2)

```
SQL> SELECT object_name
  2  FROM dba_objects
  3  WHERE object_name LIKE '%LOGSTDBY%';
```

```
DBA_LOGSTDBY_EVENTS
DBA_LOGSTDBY_HISTORY
DBA_LOGSTDBY_LOG
DBA_LOGSTDBY_NOT_UNIQUE
DBA_LOGSTDBY_PARAMETERS
DBA_LOGSTDBY_PROGRESS
DBA_LOGSTDBY_SKIP
DBA_LOGSTDBY_SKIP_TRANSACTION
DBA_LOGSTDBY_UNSUPPORTED
DBMS_INTERNAL_LOGSTDBY
DBMS_LOGSTDBY
DBMS_LOGSTDBY_LIB
GV$LOGSTDBY
GV$LOGSTDBY_PROCESS
GV$LOGSTDBY_PROGRESS
GV$LOGSTDBY_STATE
GV$LOGSTDBY_STATS
GV$LOGSTDBY_TRANSACTION
GV_$LOGSTDBY
GV_$LOGSTDBY_PROCESS
GV_$LOGSTDBY_PROGRESS
GV_$LOGSTDBY_STATE
GV_$LOGSTDBY_STATS
GV_$LOGSTDBY_TRANSACTION
LOGSTDBY$APPLY_MILESTONE
LOGSTDBY$APPLY_PROGRESS
LOGSTDBY$EVENTS
LOGSTDBY$EVENTS_IND
LOGSTDBY$HISTORY
LOGSTDBY$PARAMETERS
LOGSTDBY$PLSQL
LOGSTDBY$SCN
LOGSTDBY$SKIP
LOGSTDBY$SKIP_SUPPORT
LOGSTDBY$SKIP_TRANSACTION
LOGSTDBY_LOG
LOGSTDBY_SUPPORT
LOGSTDBY_UNSUPPORTED_TABLES
V$LOGSTDBY
V$LOGSTDBY_PROCESS
V$LOGSTDBY_PROGRESS
V$LOGSTDBY_STATE
V$LOGSTDBY_STATS
V$LOGSTDBY_TRANSACTION
V_$LOGSTDBY
V_$LOGSTDBY_PROCESS
V_$LOGSTDBY_PROGRESS
V_$LOGSTDBY_STATE
V_$LOGSTDBY_STATS
V_$LOGSTDBY_TRANSACTION
```

# Oracle Built-in Packages

```
SQL> SELECT DISTINCT object_name, overload
  2  FROM all_arguments
  3  WHERE package_name = 'DBMS_DRS‘
  4  ORDER BY 1,2;
```

```
OBJECT_NAME                    OVERLOAD
------------------------------ --------
CANCEL_REQUEST
DELETE_REQUEST
DG_BROKER_INFO
DO_CONTROL                     1
DO_CONTROL                     2
DO_CONTROL_RAW                 1
DO_CONTROL_RAW                 2
DUMP_META
GETMIV
GET_PROPERTY
GET_PROPERTY_OBJ
GET_RESPONSE
GET_RESPONSE_RAW
PING
READYTOFAILOVER
SLEEP
STATECHANGERECORDED
```

# Identifying Destination Settings

```
SQL> SELECT DEST_ID,VALID_TYPE,VALID_ROLE,VALID_NOW
  2  FROM V$ARCHIVE_DEST;
DEST_ID VALID_TYPE      VALID_ROLE   VALID_NOW
------- --------------- ------------ --------------
      1 ONLINE_LOGFILE  ALL_ROLES    YES
      2 STANDBY_LOGFILE STANDBY_ROLE YES
      3 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      4 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      5 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      6 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      7 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      8 ALL_LOGFILES    ALL_ROLES    UNKNOWN
      9 ALL_LOGFILES    ALL_ROLES    UNKNOWN
     10 ALL_LOGFILES    ALL_ROLES    UNKNOWN
     11 ALL_LOGFILES    ALL_ROLES    YES
11 rows selected.
```